JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Hate Speech Detection using Machine Learning

[1]Satrughan Kumar, [2]Arun Kumar, [3]Sandeep Kumar

[1]B. Tech Student, [2]B. Tech Student, [3]Associate Professor
Department of Computer Science and Engineering
[1]Sharda University, Greater Noida (U.P), India

Abstract: The increasing prevalence of social media and information sharing has undoubtedly benefited society in numerous ways. However, it has also brought about significant challenges, particularly concerning the proliferation of hate speech messages. To address this pressing issue within the realm of social media, recent studies have harnessed various feature engineering techniques and machine learning algorithms to automatically identify and combat hate speech across different datasets. To date, there has been no comprehensive study that systematically compares the myriad feature engineering techniques and machine learning algorithms, aiming to determine which combinations yield superior results on a commonly accessible dataset. As a response to this research gap, our paper sets out to conduct such a comparative analysis. We seek to assess the performance of three distinct feature engineering techniques in conjunction with eight diverse machine learning algorithms. Our experimental findings demonstrate that when employing bigram features in combination with the Decision Tree (DT) algorithm, the highest overall accuracy, reaching 89%, is achieved. This outcome suggests that this specific approach holds significant promise in the battle against hate speech The implications of our study extend beyond the research community. It holds practical significance by providing a foundational understanding of hate speech detection and can serve as a benchmark for future investigations in this domain. Furthermore, the insights derived from these comparative analyses will serve as state-of-the-art techniques for assessing and guiding future research endeavours focused on automated text classification techniques.

Index Terms – Decision Tree, Support Vector Machine (SVM), Natural Language Processing (NLP).

I. INTRODUCTION: -

In recent years, the proliferation of hate speech in both offline and online communication has become a pressing concern. Online platforms, especially social media, have become significant breeding grounds and vectors for the dissemination of hateful content, contributing to an alarming increase in hate crimes. Recent surveys have linked the surge in online hate speech to events such as when election held in India some leaders charge someone face, race, and caste., and terrorist attacks in Israil [4]. To address the harmful consequences of hate speech, various measures, including legislation, have been implemented by the European Union Commission. Notably, the Commission has compelled social media networks to sign an EU hate speech code, obliging them to swiftly remove hate speech content within 24 hours [1]. Nevertheless, the manual process of identifying and eliminating hate speech content is labour-intensive and time-consuming. Given the pervasive nature of hate speech on the internet, there is a strong incentive to develop automated hate speech detection systems. These studies have employed diverse feature engineering techniques and machine learning (ML) algorithms to classify content as hate speech. However, it remains difficult to compare the performance of these approaches in classifying hate speech content. To the best of our knowledge, existing studies lack a comprehensive comparative analysis of different feature engineering techniques and ML algorithms. To address this gap, this study aims to compare three feature engineering methods and eight ML classifiers using standard hate speech datasets. In Table I, Literature survey is given which is already work in the field of hate speech detection

1.1. LITERATURE SURVEY: -

In today's digital age, the prevalence of hate speech on social media platforms has become a concerning issue. Consequently, in recent years, several researchers have turned to supervised machine learning (ML)-based text classification methods to tackle the challenge of identifying and categorizing hate speech content. These studies have explored a range of feature representation techniques, including dictionary-based [21-23], Bag-of-Words-based (BoW) [24-26], N-grams-based [27-29], TF-IDF-based [30, 31], and Deep Learning-based [31] approaches. One noteworthy study conducted by Peter Burnap et al. [20] employed a dictionary-based strategy to detect cyber hate on Twitter. Their research utilized N-grams as a feature engineering technique to transform predefined hateful words into numeric vectors. These vectors were then input into a machine learning classifier, specifically Support Vector Machine (SVM), resulting in an F-score of up to 67%.

Stéphan Tulkens et al. [22] also adopted a dictionary-based approach for the automated identification of racism in Dutch social media. In their investigation, they leveraged word distribution over three dictionaries as features and used SVM as the classification method, achieving an F-score of 0.46.

Njagi Dennis et al. [21] pursued a machine learning-based approach to classify hate speech in web forums and blogs. Their methodology involved the creation of a master feature vector using a dictionary-based approach. This vector was constructed based on sentiment expressions with semantic and subjectivity features aimed at detecting hate speech. The authors then fed the master feature vector into a rule-based classifier, achieving a precision metric of 73%.

While the combination of dictionary-based and ML approaches demonstrated promise, a significant drawback is the requirement for extensive domain-specific dictionaries. In response, many researchers have explored the Bag-of-Words (BoW) approach, which is conceptually like the dictionary-based method but derives word features from training data rather than predefined dictionaries.

Edel Greevy et al. [23] adopted a supervised ML approach for the classification of racist text. They employed bigram feature extraction to convert raw text into numeric vectors and used these bigram features in conjunction with the BoW technique. Their chosen classifier was DT, resulting in an impressive 89% accuracy rate.

Irene Kwok et al. [24] addressed the automatic detection of racism against Black individuals within the Twitter community using an ML-based approach. They utilized unigrams with the Bow-based technique to generate numeric vectors and applied the Naïve Bayes classifier. Their experimental results yielded a maximum accuracy rate of 76%.

Sanjana Sharma et al. [25] focused on classifying hate speech on Twitter and employed BoW features. These numeric vectors were fed into the Naïve Bayes classifier, with their experimental outcomes showing a maximum accuracy rate of 73%.

The BoW technique has proven effective for accuracy in social network text classification, but it has limitations as it ignores word order, leading to potential misclassifications when words have different meanings in distinct contexts. In response to this challenge, researchers have proposed an N-grams-based approach [7].

2. METHODOLOGY: -

In this section, system for classifying tweets into three distinct categories: "hate speech," "offensive but not hate speech," and "neither hate speech nor offensive speech." The complete research methodology is illustrated in Figure 1, comprising six essential steps: data collection, data preprocessing, feature engineering, data splitting, construction of the classification model, and evaluation of the classification model. Each of these steps will be comprehensively explained in the following sections.

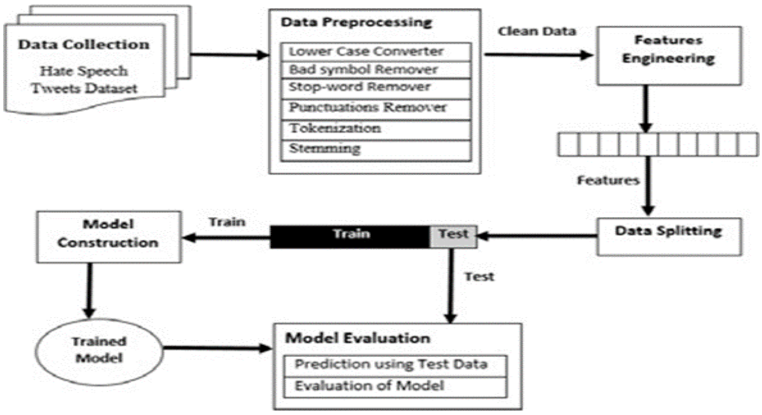


Figure 1. System Overviews [7]

A. Data Collection: -

In our research study, we utilized a publicly available dataset containing hate speech tweets, which had been compiled and labeled by CrowdFlower. This dataset categorizes tweets into three distinct classes: "hate speech," "not offensive," and "offensive but not hate speech." The dataset comprises a total of 26,000 tweets, with the following distribution: 16% of the tweets fall into the "hate speech" category, 50% into the "not offensive" category, and the remaining 33% belong to the "offensive but not hate speech" category.

B. Text Preprocessing: -

Numerous research studies have highlighted the efficacy of text preprocessing in enhancing classification outcomes [33]. In our dataset, we implemented various preprocessing techniques aimed at eliminating noisy and uninformative features from the tweets. As part of this preprocessing, we converted all tweets to lowercase. Additionally, we employed pattern matching techniques to eliminate URLs, usernames, white spaces, hashtags, punctuation, and stop words from the collected tweets. Furthermore, we performed tokenization and stemming on the preprocessed tweets. Tokenization involved breaking down each individual tweet into tokens or words, and then the Porter stemmer was utilized to reduce words to their root forms. For instance, it transformed words like "offended" into "offend" using the Porter stemmer.

C. Data Splitting: -

Table II provides insights into the distribution of classes within the entire dataset, both before and after the data split, which results in a training set and a test set. We applied an 80-20 ratio for this data split, allocating 80% of the preprocessed data to the training dataset and reserving 20% for the test dataset. The training dataset serves as the foundation for training the classification model, allowing it to acquire the necessary classification rules. Subsequently, the test dataset is employed to assess the effectiveness of the classification model. In this, there are some specific operations are taking place to divide the dataset into the ratio of eighty and twenty.

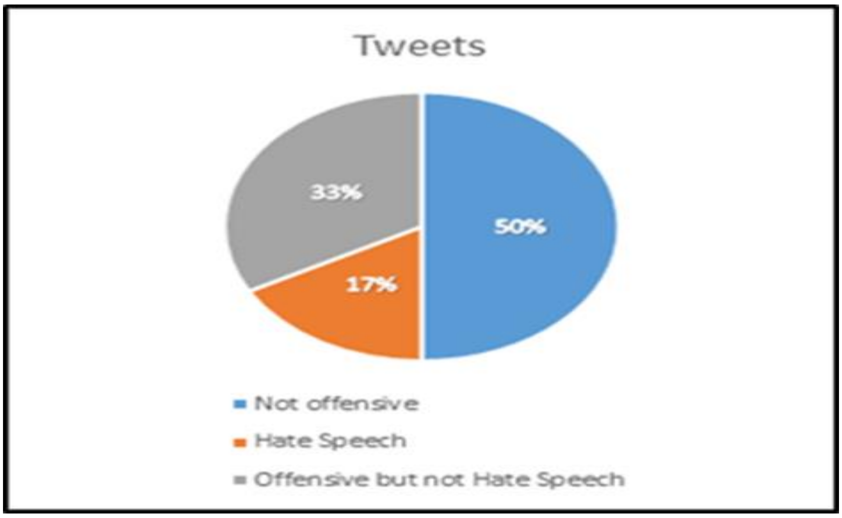


Figure. 2. Class Wise Distribution

TABLE I. DETAILS OF DATA SPLIT

	Class	**Total Instances**	**Traininginstances**	**Testing instances**
0	Hate Speech	8489	1909	490
1	Not offensive	7574	5815	1459
2	Offensive but not Hate Speech	5836	3883	953
	Total	**21410**	**1607**	**2902**

D. Machine Learning Models: -

The "No Free Lunch Theorem," as stated in reference [34], asserts that no single classifier can universally outperform all others across diverse datasets. Consequently, it is advisable to employ a variety of classifiers on a consolidated feature vector to assess which one yields superior results. Considering this, we have opted for eight distinct classifiers: Naive Bayes (NB) [12], Support Vector Machine (SVM) [14], k-Nearest Neighbors (KNN) [15], Decision Tree (DT) [16], Random Forest (RF) [13], AdaBoost [17], Multilayer Perceptron (MLP) [18], and Logistic Regression (LR) [19].

E. Classifier Evaluation: -

In this phase, the constructed classifier is tasked with predicting the classification of unlabelled text, specifically categorizing it as "hate speech," "offensive but not hate speech," or "neither hate speech nor offensive speech" using a test dataset. To evaluate the performance of the classifier, we assess its effectiveness by computing the following metrics: True Negatives (TN), False Positives (FP), False Negatives (FN), and True Positives (TP). These four values collectively form a confusion matrix, as illustrated in Figure 3. Various performance measures are employed to gauge the classifier's performance. Here are some common performance metrics used in text categorization, with a brief explanation of each:

1. Precision (Positive Predictive Value): Precision, also known as the positive predicted value, quantifies the proportion of predictive positives that are genuinely positive. It is computed as follows (see equation 1):
2. Precision = TP / (TP + FP) (1)
3. Recall: Recall measures the proportion of actual positive instances that the classifier correctly predicts as positive. It is calculated as follows (refer to equation 2):

   Recall = TP / (TP + FN) (2)
4. F-Measure: The F-Measure is the harmonic mean of precision and recall, as indicated in Equation 3. The standard F-Measure (F1) assigns equal importance to both precision and recall. It can be computed using the following formula (see equation 3):

F-Measure = 2 * (Precision * Recall) / (Precision + Recall) (3)

5. Accuracy: Accuracy represents the number of correctly classified instances, including both true positives and true negatives. It is calculated as follows (refer to equation 4):

   Accuracy = (TP + TN) / (TP + FP + TN + FN) (4)

For a more comprehensive understanding of these performance metrics, additional details can be found in reference [35].

3. RESULTS: -

This section provides a comprehensive overview of the results obtained from 24 different analyses. These tables showcase the performance of various feature representations and classification techniques within the experimental setup.

Across all 24 analyses, the lowest precision (0.58), recall (0.57), accuracy (57%), and F-measure (0.47) were found in the Multilayer Perceptron (MLP) and k-Nearest neighbours (KNN) classifiers when using TFIDF feature representation with bigram features.

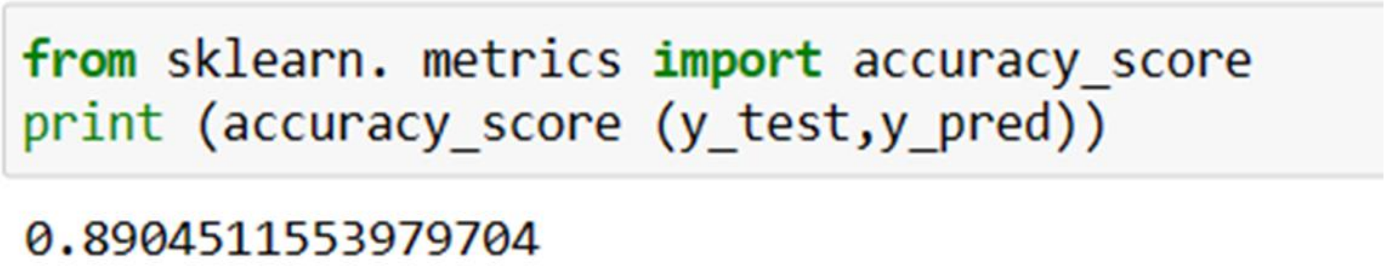
On the other hand, the highest recall (0.89), precision (0.77), accuracy (89%), and F-measure (0.77) were achieved by the Decision Tree (DT) classifier when using TFIDF feature representation with bigram features. Notably, among the different feature representations, bigram features combined with TFIDF yielded the best overall performance when compared to Word2vec and Doc2vec. Nevertheless, there was only a slight margin between the results obtained with bigram and Doc2vec representations.

In the realm of text classification models, the DT classifier stood out as the top performer among the eight classifiers assessed. However, the AdaBoost and Random Forest (RF) classifiers yielded results that were somewhat below those of the DT, yet notably better than the results produced by Logistic Regression (LR), Support Vector Machine (SVM), Naive Bayes (NB), k-Nearest Neighbours (KNN), and Multilayer Perceptron (MLP) classifier [46].

TABLE Il. ACCURACY OF ALL 07 ANALYSIS

Features	**LR**	**NB**	**RF**	**SVM**	**KNN**	**AdaBoost**	**DT**
Bigram	0.75	0.73	0.75	**0.79**	**0.57**	0.78	**0.89**
Word2vec	0.72	0.67	0.68	0.73	**0.61**	0.68	0.75
Doc2vec	0.72	**0.62**	0.67	0.72	0.65	0.67	0.73

Accuracy through Decision Tree algorithm: -



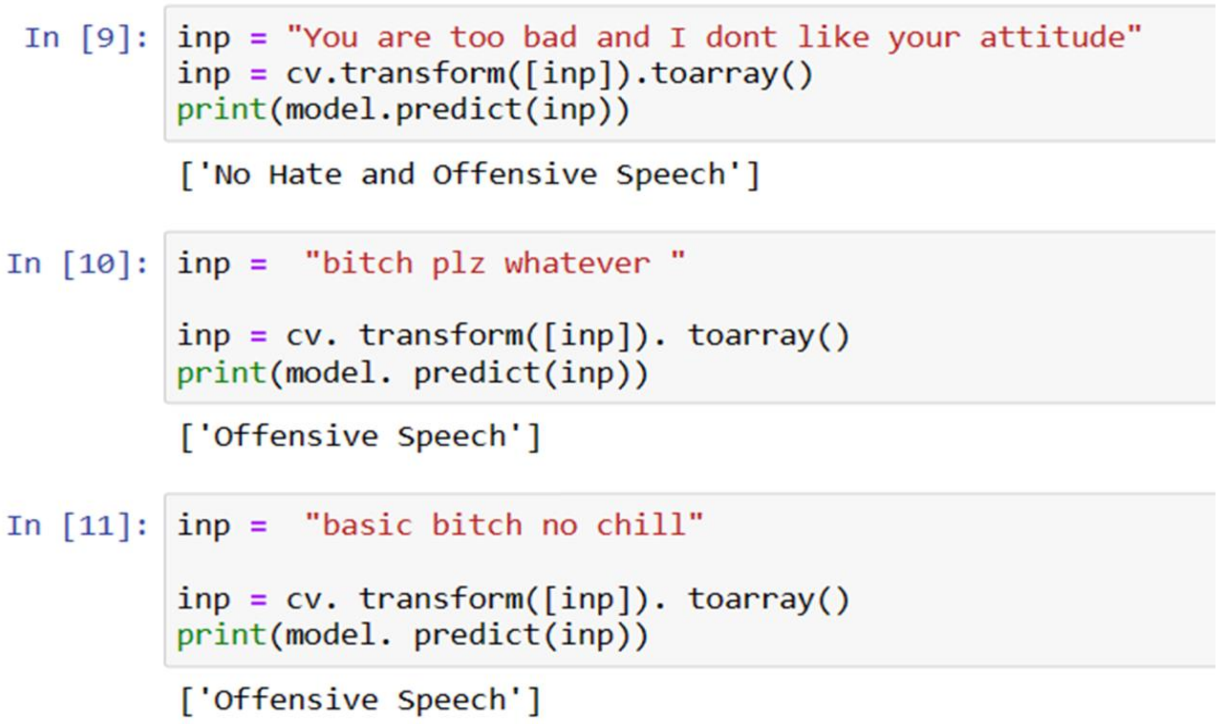
```
from sklearn. metrics import accuracy_score
print (accuracy_score (y_test,y_pred))

0.8904511553979704
```

Figure. 3. Accuracy of Decision Tree Algorithm.

Here, it shown the 0.890 accuracy through Decision Tree Algorithm, and logistic regression shown 0.75, and all algorithms accuracy mentioned above, the second highest accuracy given by the Support Vector Machine (SVM) is 0.79 accuracy on the same dataset are using for all the algorithms.

The result finally given, when comment is passed the “you are too bad and I don’t like your attitude” it shown the No Hate and offensive speech, but when a new sentence or word pass, “bitch plz whatever” it gave the “offensive speech”. When other sentence pass like, “you are so good”, it gave the “No offensive speech”. Through it, it can able to check the hate comments.



```
In [9]:  inp = "You are too bad and I dont like your attitude"
         inp = cv.transform([inp]).toarray()
         print(model.predict(inp))

         ['No Hate and Offensive Speech']

In [10]: inp =  "bitch plz whatever "

         inp = cv. transform([inp]). toarray()
         print(model. predict(inp))

         ['Offensive Speech']

In [11]: inp =  "basic bitch no chill"

         inp = cv. transform([inp]). toarray()
         print(model. predict(inp))

         ['Offensive Speech']
```

Figure. 4. Hate comment detection.

In addition, let us examine Fig. 5 and Fig. 6, which depict the confusion matrices for the best-performing analyses. In Fig. 4, we observe the confusion matrix for DT classifiers using bigram with TFIDF features. It is evident that out of the 490 tweets classified as hate speech, only 155 were correctly identified, while 335 were misclassified. Among these 335 misclassified instances, 54 were erroneously categorized as not offensive, and 281 were wrongly labelled as offensive language but not hate speech. Moving on to the 1459 instances in the second class, 1427 tweets were accurately classified as not offensive speech, but 32 instances were misclassified. Five of them were incorrectly labelled as hate speech, and 27 were falsely categorized as offensive language but not hate speech. The remaining 953 instances from the 2902-test set belong to the offensive language but not hate speech class. In this case, the DT classifier correctly identified 698 tweets as offensive language but not hate speech, while 122 and 133 instances were misclassified as hate speech and not offensive speech, respectively.

Now, Fig. 6 illustrates the confusion matrix for the SVM classifier when using bigram with TFIDF features. It is worth noting that the overall performance of the SVM classifier is lower compared to the SVM classifier under these conditions. SVM performance is mainly satisfactory for the offensive language but not hate speech.

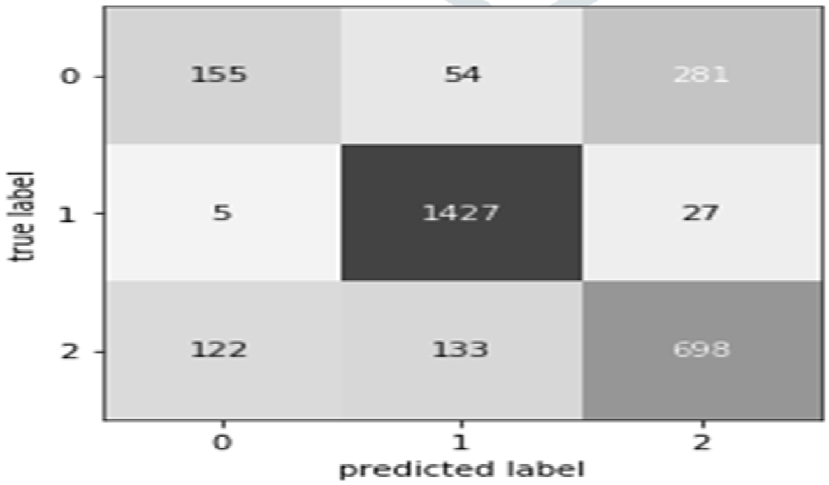


Figure. 5. Confusion Matrix (Features: Bigram, Classifier, D.T).

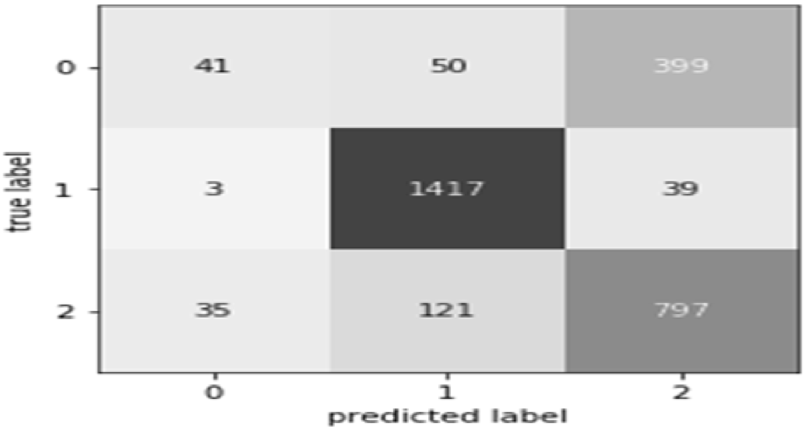


Figure. 6 Confusion Matix (Feature Bigram (TFIDF), Classifier, SVM)

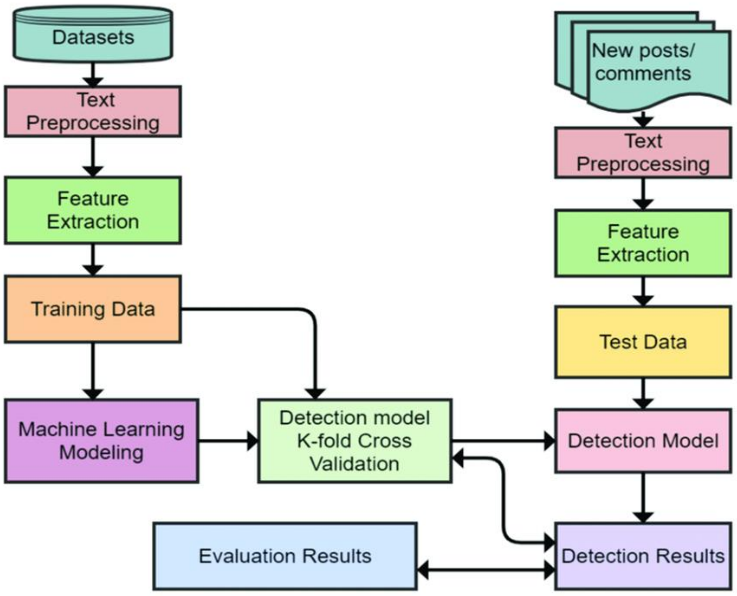


Figure. 7. Hate Speech Detection Architecture.

In this architecture, firstly database, which contains the posts or offensives, no offensive or partially offensive, it checks the new posts and comments. It takes text processing and Feature extraction then train and test the data after it, it needs to modeling or check the speech or comments are offensive or not, Support Vector Machines, Decision Tree, Linear Regression algorithms techniques are used. After it, it can be able to detect the offensive, No hate and offensive, No offensive and partially offensive detection. At the last it can evaluation the results.

3.1. DISCUSSION: -

In this experimental study, we systematically assessed the performance of eight distinct classifiers across three distinct feature engineering techniques. This comprehensive analysis resulted in a total of 24 distinct evaluations conducted on a hate speech dataset comprising three different classes. Our findings revealed that the SVM (Support Vector Machine) algorithm, particularly when combined with the bigram feature extraction technique and TFIDF (Term Frequency-Inverse Document Frequency) feature engineering, delivered the most promising and superior results. For a detailed exploration of the theoretical underpinnings and implications of these results, we invite you to delve into the subsequent sections of this report.

A. Feature Engineering

Feature engineering plays a crucial role in text classification, and this study underlines its significance. We conducted a comparative analysis of three distinct feature extraction techniques: Bigram with TFIDF, Word2Vec, and Doc2Vec. The empirical results underscored a clear distinction in their performance. Among these techniques, Bigram with TFIDF emerged as the top performer, while Word2Vec and Doc2Vec lagged in terms of efficacy. The superior performance of Bigram with TFIDF can be attributed to its ability to preserve the sequence of words, a feature that sets it apart from Word2Vec and Doc2Vec [36]. Furthermore, this preference aligns with previous research, which has consistently demonstrated the advantages of the TFIDF representation technique over binary and term frequency representations [6]. These findings emphasize the critical role of feature engineering in text classification, particularly when seeking the most effective methods to enhance classification accuracy.

B. Machine Learning Classifier

Numerous studies have consistently shown that there isn't a one-size-fits-all machine learning algorithm that excels across all types of data. Therefore, it's imperative to conduct a thorough comparison of multiple machine learning algorithms to pinpoint the one that performs optimally on a given dataset. In our specific study, we evaluated the performance of eight different machine learning algorithms, as detailed in Section 3.E, under the heading "ML Models." Our experimental findings demonstrated that SVM (Support Vector Machine) and AdaBoost classifiers exhibited the most impressive performance. This outcome can be attributed to SVM's capability to use threshold functions for data separation, focusing on margin rather than the number of features, making it less dependent on feature quantity [7, 15]. Additionally, SVM's ability to handle non-linear data effectively through kernel functions contributed to its success.

As for AdaBoost, its strength lies in its adaptive learning algorithms, which iteratively improve classification rules [39] and its emphasis on reducing training errors. Although Random Forest (RF) and Logistic Regression (LR) classifiers achieved slightly lower performance compared to SVM and AdaBoost, they outperformed Naive Bayes (NB), Decision Trees (DT), k-Nearest Neighbours (KNN), and Multilayer Perceptron (MLP) classifiers.

C. Class wise Performance

In this model "hate speech," "offensive but not hate speech," and "neither hate speech nor offensive speech." The results revealed that all features and classifiers performed admirably when distinguishing between the "offensive but not hate speech" and "neither hate speech nor offensive speech" classes. However, in our experimental findings, we observed that the 24 different combinations of features and classifiers yielded the lowest performance when tasked with classifying instances as "hate speech. “Analysing Table I, it becomes apparent that the "Hate Speech" class had the fewest training instances compared to the other classes. Nevertheless, the primary reason for the misclassification of instances within the "Hate Speech" class, as illustrated in Fig. 3 and Fig. 4, appeared to be the presence of different bigram words that were more frequent in other classes, rather than in the "Hate Speech" class. For instance, bigrams like "lame nigga," "white trash," and "bitch made" were notably more prevalent in the "Offensive but not Hate Speech" class in contrast to the "Hate Speech" class. This suggests that the classifier might have learned relatively weaker rules for distinguishing "Hate Speech" from the other classes. The overlapping use of certain bigram words between the "Hate Speech" and "Offensive but not Hate Speech" classes likely led to misclassifications, emphasizing the need for more robust and discriminative rules or features to enhance the accurate classification of "Hate Speech."

3.2. CONCLUSION: -

This research harnessed automated text classification techniques to identify hate speech messages. Furthermore, the study conducted a comprehensive comparison of three feature engineering techniques and eight machine learning (ML) algorithms for the purpose of categorizing hate speech messages. The experimental outcomes notably demonstrated the superior performance of bigram features when represented through TFIDF, surpassing the effectiveness of the word2Vec and Doc2Vec feature engineering techniques. Among the ML algorithms, SVM and RF exhibited superior results compared to LR, NB, KNN, DT, AdaBoost, and MLP. It is worth noting that KNN yielded the lowest performance.

The insights derived from this research hold significant practical importance, as they establish a foundational benchmark for future investigations in the domain of automatic hate speech detection. Furthermore, this study contributes scientifically by presenting experimental results using multiple scientific measures for automatic text classification.

However, it's important to acknowledge two key limitations of our work. Firstly, the proposed ML model displayed inefficiency in real-time prediction accuracy for the data. Secondly, it limited its classification to three distinct classes without the capability to gauge the severity of the hate speech messages. Our future objectives revolve around refining the proposed ML model to enable the prediction of message severity. Additionally, we aim to enhance the classification performance using two strategies. First, we will explore and evaluate lexicon-based techniques by comparing them with current state-of-the-art results. Second, we intend to expand our dataset with more instances to facilitate more effective learning of classification rules.

4. FUTURE SCOPE: -

In the future point of views, it is most beneficial for the social media company and crime branch of the police, if any one delivered its offensive speech and by the using of this model social media company can ban the video which is uploaded by the people and it can control the crime. And on the other hand, Crime branch can urge the people and social media operator, this is fake comments which is show at the name of someone. Hence, with the help of this model we can control the societal issues and hate which is emerge in the society.

6. REFERENCES

[1] Hern, A., Facebook, YouTube, Twitter, and Microsoft sign the EU hate speech code. The Guardian, 2016. 31.

[2] Rosa, J., and Y. Bonilla, Deprovincializing Trump, decolonizing diversity, and unsettling anthropology. American Ethnologist, 2017. 44(2): p. 201-208.

[3] Travis, A., Anti-Muslim hate crime surges after Manchester and London Bridge attacks. The Guardian, 2017.

[4] MacAvaney, S., et al., Hate speech detection: Challenges and solutions. PloS one, 2019. 14(8): p. e0221152.

[5] Fortuna, P. and S. Nunes, A survey on automatic detection of hate speech in text. ACM Computing Surveys (CSUR), 2018. 51(4): p. Mujtaba, G., et al., Prediction of cause of death from forensic autopsy reports using text classification techniques: A comparative study. Journal of forensic and legal medicine, 2018. 57: p. 41-50.

[6] Cavnar, W.B. and J.M. Trenkle. N-gram-based text categorization. in Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval. 1994. Citeseer.

[7] Ramos, J. Using tf-idf to determine word relevance in document queries. in Proceedings of the first instructional conference on machine learning. 2003. Piscataway, NJ.

[8] Mikolov, T., et al. Distributed representations of words and phrases and their compositionality. in Advances in neural information processing systems. 2013.

[9] Le, Q. and T. Mikolov. Distributed representations of sentences and documents. in International conference on machine learning. 2014.

[10] Kotsiantis, S.B., I.D. Zaharakis, and P.E. Pintelas, Machine learning: a review of classification and combining techniques. Artificial Intelligence Review, 2006. 26(3): p. 159-190.

[11] Lewis, D.D. Naive (Bayes) at forty: The independence assumption in information retrieval. in European conference on machine learning. 1998. Springer.

[12] Xu, B., et al., An Improved Random Forest Classifier for Text Categorization. JCP, 2012. 7(12): p. 2913-2920.

[13] Joachims, T. Text categorization with support vector machines: Learning with many relevant features. in European conference on machine learning. 1998. Springer.

[14] Zhang, M.-L. and Z.-H. Zhou, A k-nearest neighbor based algorithm for multi-label classification. GrC, 2005. 5: p. 718-721.

[15] Abacha, A.B., et al., Text mining for pharmacovigilance: Using machine learning for drug name recognition and drug–drug interaction extraction and classification. Journal of biomedical informatics, 2015. 58: p. 122- 132.

[16] Ying, C., et al., Advance and prospects of AdaBoost algorithm. Acta Automatica Sinica, 2013. 39(6): p. 745-758.

[17] Gardner, M.W. and S. Dorling, Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmospheric environment, 1998. 32(14-15): p. 2627-2636.

[18] Wenando, F.A., T.B. Adji, and I. Ardiyanto, Text classification to detect student level of understanding in prior knowledge activation process. Advanced Science Letters, 2017. 23(3): p. 2285-2287.

[19] Burnap, P. and M.L. Williams, Us and them: identifying cyber hate on Twitter across multiple protected characteristics. EPJ Data Science, 2016. 5(1): p. 11.

[20] Gitari, N.D., et al., A lexicon-based approach for hate speech detection. International Journal of Multimedia and Ubiquitous Engineering, 2015. 10(4): p. 215-230.

[21] Tulkens, S., et al., A dictionary-based approach to racism detection in dutch social media. arXiv preprint arXiv:1608.08738, 2016.

[22] Greevy, E. and A.F. Smeaton. Classifying racist texts using a support vector machine. in Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. 2004. ACM.

[23] Kwok, I. and Y. Wang. Locate the hate: Detecting tweets against blacks. in Twenty-seventh AAAI conference on artificial intelligence. 2013.

[24] Sharma, S., S. Agrawal, and M. Shrivastava, Degree based classification of harmful speech using twitter data. arXiv preprint arXiv:1806.04197, 2018.

[25] Malmasi, S. and M. Zampieri, Detecting hate speech in social media. arXiv preprint arXiv:1712.06427, 2017.

[26] Nobata, C., et al. Abusive language detection in online user content. in Proceedings of the 25th international conference on world wide web. 2016. International World Wide Web Conferences Steering Committee.

[27] Waseem, Z. and D. Hovy. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. in Proceedings of the NAACL student research workshop. 2016.

[28] Dinakar, K., R. Reichart, and H. Lieberman. Modeling the detection of textual cyberbullying. in fifth international AAAI conference on weblogs and social media 2011.

[29] Liu, S. and T. Forss. Combining N-gram based Similarity Analysis with Sentiment Analysis in Web Content Classification. in KDIR. 2014.

[30] Köffer, S., et al., Discussing the value of automatic hate speech detection in online debates. Multikonferenz Wirtschaftsinformatik (MKWI 2018): Data Driven X-Turning Data in Value, Leuphana, Germany, 2018.

[31] Chen, Y., Detecting offensive language in social medias for protection of adolescent online safety. 2011.

[32] Shaikh, S. and S.M. Doudpotta, Aspects Based Opinion Mining for Teacher and Course Evaluation. Sukkur IBA Journal of Computing and Mathematical Sciences, 2019. 3(1): p. 34-43.

[33] Ho, Y.-C. and D.L. Pepyne, Simple explanation of the no-free-lunch theorem and its implications. Journal of optimization theory and applications, 2002. 115(3): p. 549-570.

[34] Seliya, N., T.M. Khoshgoftaar, and J. Van Hulse. A study on the relationships of classifier performance metrics. in 2009 21st IEEE international conference on tools with artificial intelligence. 2009. IEEE.

[35] Chaudhari, U.V. and M. Picheny, Matching criteria for vocabulary- independent search. IEEE Transactions on Audio, Speech, and Language Processing, 2012. 20(5): p. 1633-1643.

[36] Li, Y. and T. Yang, Word embedding for understanding natural language: a survey, in Guide to Big Data Applications. 2018, Springer.
p. 83-104.

[37] Wang, Y., et al. Comparisons and selections of features and classifiers for short text classification. in IOP Conference Series: Materials Science and Engineering. 2017. IOP Publishing.

[38] Schapire, R.E., The boosting approach to machine learning: An overview, in Nonlinear estimation and classification. 2003, Springer. p. 149-171.

[39] Xu, B., Y. Ye, and L. Nie. An improved random forest classifier for image classification. in 2012 IEEE International Conference on Information and Automation. 2012. IEEE.

[40] Eftekhar, B., et al., Comparison of artificial neural network and logistic regression models for prediction of mortality in head trauma based on initial clinical data. BMC medical informatics and decision making, 2005. 5(1): p. 3.

[41] Dreiseitl, S., et al., A comparison of machine learning methods for the diagnosis of pigmented skin lesions. Journal of biomedical informatics, 2001. 34(1): p. 28-36.

[42] Singh, P.K. and M.S. Husain, Methodological study of opinion mining and sentiment analysis techniques. International Journal on Soft Computing, 2014. 5(1): p. 11.

[43] Bhatia, N., Survey of nearest neighbor techniques. arXiv preprint arXiv:1007.0085, 2010.

[44] Sigurbergsson, G. I., & Derczynski, L. (2019). Offensive language and hate speech detection for Danish. arXiv preprint arXiv:1908.04531.

[45] Schmidt, A., & Wiegand, M. (2017, April). A survey on hate speech detection using natural language processing. In Proceedings of the Fifth International workshop on natural language processing for social media (pp. 1-10).

[46] (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 11, No. 8, 2020.